



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/766,758

01/27/2004

Robert Jensen

021751-000710US

2259

20350

7590

09/08/2006

TOWNSEND AND TOWNSEND AND CREW, LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

EXAMINER

VAUTROT, DENNIS L

ART UNIT

PAPER NUMBER

2167

DATE MAILED: 09/08/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

10/766,758

Applicant(s)

JENSEN ET AL.

Examiner

Dennis L. Vautrot

Art Unit

2167

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 27 January 2004.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-20 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-20 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 27 January 2004 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____

DETAILED ACTION

Drawings

1. The drawings are objected to because the first box in FIG. 3 refers to item 200, whereas the specification refers to item 300. Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Objections

2. Claim 13 is objected to because of the following informalities: In line 11, "...instance of the third object" is referred to, with no antecedent basis for the third object. Appropriate correction is required.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 1, 6, 7, 14, 17, and 18 are rejected under 35 U.S.C. 102(e) as being anticipated by **Szabo et al.** (6,768,486).
5. Regarding claim 1, **Szabo et al.** (hereinafter **Szabo**) teaches a method for a computer system comprises: opening a first file [derived object] describing a first object in an object environment; determining a reference for a second object [base object], wherein the second object includes a plurality of attributes [basic description] (See column 14, lines 31-32 "Derived Object – includes a list of modifier objects and a pointer to the base object." And see column 16, lines 7-9 "Specifically, base object 810 provides

reference; (See column 16, lines 8-9 "The basic description is then provided to the object space modifier 820 in the modifier stack 880."); opening the second file describing the second object [base object] in the object environment; determining a modified value [modifiers] for an attribute from the plurality of attributes for the second object; and including in the first file the reference for the second object [base object] and the modified value [modifiers] for the attribute; wherein the second object is not stored in the first file. (See column 15, lines 9 – 14 "Derived object 870 maintains a modifier stack 880. The modifier stack 880 includes a list of modifiers..., a transform 830, and a pointer to a base object 810. the derived object 870 maintains a reference to base object 810, a reference to transform 830, and a list of modifiers." Because the reference is stored in the file, the actual second object is not stored in the first file.)

6. Regarding claim 6, **Szabo** additionally discloses opening a third file describing a third object [multiple objects] in an object environment, wherein the third file includes a reference [pointer] to the second object [base object] and includes a override value [modifier] for the attribute (See column 11, lines 51-53 "The reference allows a scene to be rendered and allows for the use of modifiers by multiple objects." And see column 14, lines 31-32 "Derived Object – includes a list of modifier objects and a pointer to the base object." Allowing for use by multiple objects would include a third object, or any number of objects.); determining the reference [pointer] for the second object [base object] in response to the third file (See column 14, lines 31-32 "Derived Object – includes a list of modifier objects and a pointer to the base object." And see column 16,

lines 7-9 "Specifically, base object 810 provides the basic description of the object."); receiving a second file [description is then provided] describing the second object [base object] from the storage system in response to the reference; (See column 16, lines 8-9 "The basic description is then provided to the object space modifier 820 in the modifier stack 880."); opening the second file [modifier stack] describing the second object [base object] in the object environment; and modifying the value [basic description] for the attribute from a default value to the override value in response to the third file. (See column 15, lines 9 – 14 "Derived object 870 maintains a modifier stack 880. The modifier stack 880 includes a list of modifiers..., a transform 830, and a pointer to a base object 810." And see column 16, lines 10 – 11 "Object space modifier 820 modifies this basic description to generate a modified description.")

7. Regarding claim 7, **Szabo** additionally discloses modifying the second file to include an additional attribute [changes] of the second [base] object (See column 11, lines 53-55 "The reference allows changes in the base object to be broadcast to any objects that are referring to that base object." Adding an additional attribute is one of many changes that could be incorporated.); reopening the first file describing the first object in the object environment; determining the reference [pointer] for the second object [base object] in response to the third file (See column 14, lines 31-32 "Derived Object – includes a list of modifier objects and a pointer to the base object." And see column 16, lines 7-9 "Specifically, base object 810 provides the basic description of the object."); receiving the second file [basic description] as modified describing the second

object [base object] as modified from the storage system in response to the reference (See column 16, lines 8-9 "The basic description is then provided to the object space modifier 820 in the modifier stack 880." Here, because the changes to the second object is propagated upon being referenced, the modified object is the one that will be received.); opening the second file as modified describing the second object [base object] as modified in the object environment; modifying a value for the additional attribute [modified description] of the second object [base object] in the object environment; (See column 15, lines 9 – 14 "Derived object 870 maintains a modifier stack 880. The modifier stack 880 includes a list of modifiers... a transform 830, and a pointer to a base object 810." And see column 16, lines 10 – 11 "Object space modifier 820 modifies this basic description to generate a modified description."); and including in the first file the value for the additional attribute (See column 16, lines 7 - 8 "Specifically, base object 810 provides the basic description of the object." Here, the basis description would now include the additional attribute as it was propagated when subsequently referenced.)

8. Regarding claim 14, **Szabo** discloses a computer program product for a computer system including a processor coupled to a server comprises code that directs the processor to allow a user to create a first object [derived object] in an object environment (See column 15, lines 8-9 "The derived object instance is created when a user creates an object in a scene."); code that directs the processor to determine a reference for a second object [base object] in the server, wherein the second object

Art Unit: 2167

includes a plurality of attributes [basic description] (See column 14, lines 31-32 "Derived Object – includes a list of modifier objects and a pointer to the base object." And see column 16, lines 7-9 "Specifically, base object 810 provides the basic description of the object."); code that directs the processor to create an instance of the second object in the object environment (See column 15, lines 8-9 "The derived object instance is created when a user creates an object in a scene."); code that directs the processor to determine a modified value [modifiers] for an attribute from the plurality of attributes for the second object [base object] (See column 15, lines 9 – 14 "Derived object 870 maintains a modifier stack 880. The modifier stack 880 includes a list of modifiers..., a transform 830, and a pointer to a base object 810. the derived object 870 maintains a reference to base object 810, a reference to transform 830, and a list of modifiers."); and code that directs the processor to override a default value [modifies] for the attribute [description] with the modified value (See column 16, lines "Object space modifier 820 modifies the basic description to generate a modified description."); wherein the attribute of second object stored in the server is not modified (See column 2, lines 16 – 26 "...define a base object by selecting a particular object type from a set of predefined object types and selecting a set of parameter values that are the be used to define the specific parameters of the base object...the user may define one or more modifiers or other types of components that are applied to the base object for modifying certain characteristics, properties, attributes, constraints and other parameters of the base object...the object can then be exported for use in a graphics application." Once the base object is defined, the attributes stored on the server is not modified, rather

modifiers can be applied to instances of a created object that references the base object.); and wherein the codes reside on a tangible media (See column 40, lines 56 – 58 “Such instructions may be read into main memory 706 from another computer-readable medium such as a storage device”).

9. Regarding claim 17, **Szabo** teaches code that directs the processor to create an additional instance of the second object in the object environment (See column 15, lines 7-9 “The derived object 870 is an instance of the DerivedObject class. The derived object instance is created when a user creates an object in a scene.”); code that directs the processor to determine a modified value for an attribute [modifier] from the plurality of attributes for the additional instance of the second object (See column 15, lines 9-10 “Derived object 870, maintains a modifier stack 880. The modifier stack 880 includes a list of modifiers...” Here, the value of the attribute is overridden based on that value of the modifier in the modifier stack.); and code that directs the processor to override a default value [modify the description] of the attribute for additional instance of the second object with the value (See column 16, lines “Object space modifier 820 modifies this basic description to generate a modified description.”).

10. Regarding claim 18, **Szabo** additionally discloses the modified value [modifiers] for the attribute for the instance of the second object and the modified value for the attribute for the additional instance of the second object are different. (See column 15, lines 9-10 “Derived object 870 maintains a modifier stack 880. The modifiers stack 880

includes a list of modifiers..." Because each derived object maintains its own stack, the attribute modifiers can be different for each object.)

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

12. Claims 2, 3, and 5 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Szabo** as applied to claim 1 above and further in view of **Buxton et al.** (hereinafter **Buxton**, US 5,970,252).

13. Regarding claim 2, **Szabo** teaches a method substantially as claimed. **Szabo** fails to teach the storage system comprises one of the group: network directory, asset management system, database management system. However, **Buxton** teaches the storage system comprises one of the group: network directory, asset management system, database management system. (See column 10, lines 41-45 "An ISTORE is an OLE interface which supports the creation and management of structure storage objects. Such structured storage objects provide hierarchical storage of information within a single file similar to a file system within a file." This is similar to a network directory.) A person with ordinary skill in the art would have been motivated to combine the references because they both relate to the art of modifying data objects and they

would have been motivated to include the storage system from **Buxton**, in order to maintain a persistent copy of the objects and modifiers rather than using a temporary stack, providing for a more efficient method. It is for this reason that one of ordinary skill in the art would have been motivated to include the storage system comprises one of the group: network directory, asset management system, database management system.

14. Regarding claim 3, **Szabo** teaches the reference [pointer] to the second object [base object] and the modified value for the attribute [modifiers]. (See column 15, lines 9-12 "Derived object 870 maintains a modifier stack 880. The modifier stack 880 includes a list of modifiers...and a pointer to a base object") **Szabo** fails to teach storing the first file in the storage system. However **Buxton** teaches storing the first file in the storage system. (See column 11, lines 10-13 "In addition, these templates may be stored to facilitate efficient distribution of templates to others...." Buxton teaches storing template files, whereas template files are not used in Szabo, rather a modifier stack is used.) A person with ordinary skill in the art would have been motivated to combine the references because they both relate to the art of modifying data objects and they would have been motivated to also include storing the reference and modified value, in order to facilitate a more efficient distribution of the reference and modifiers. It is for this reason that one of ordinary skill in the art would have been motivated to include storing the first file in the storage system.

15. Regarding claim 5, **Szabo** additionally discloses the first file includes a reference to third object [base object] and a override value [modifiers] for an attribute of the third object (See column 11, lines 51-53 "The reference allows a scene to be rendered and allows for the use of modifiers by multiple objects." And see column 14, lines 31-32 "Derived Object – includes a list of modifier objects and a pointer to the base object." Allowing for use by multiple objects would include a third object, or any number of objects.); and wherein the method further comprises: determining a reference [pointer] for the third object [base object] in response to the first file (See column 14, lines 31-32 "Derived Object – includes a list of modifier objects and a pointer to the base object." And see column 16, lines 7-9 "Specifically, base object 810 provides the basic description of the object."); receiving a third file [description is then provided] describing the third object from a storage system in response to the reference to the third object; (See column 16, lines 8-9 "The basic description is then provided to the object space modifier 820 in the modifier stack 880."); opening the third file [modifier stack] describing the third object [base object] in the object environment; and modifying a value [basic description] for the attribute of the third object from a default value to the override value in response to the first file. (See column 15, lines 9 – 14 "Derived object 870 maintains a modifier stack 880. The modifier stack 880 includes a list of modifiers..., a transform 830, and a pointer to a base object 810." And see column 16, lines 10 – 11 "Object space modifier 820 modifies this basic description to generate a modified description.")

16. Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Szabo** as applied to claim 1 above and further in view of **Drucker et al.** (hereinafter **Drucker**, US 5,970,252). **Szabo** teaches a method substantially as claimed. **Szabo** fails to teach geometrically coupling the first object to the second object in the object environment. (See column 10, lines 6-12 "Properties 90a, also include a geometry property that associates a geometric model for representing the object, a location property that defines a location object in the virtual world environment with or in which the particular object is positioned, a contents property that lists objects "contained" by the particular object, and an owner property that identifies a user that "owns" the particular object.") A person with ordinary skill in the art would have been motivated to combine the teachings of **Szabo** with that of **Drucker** because they both are in the art of modifying objects at run-time, and by including geometric coupling, as disclosed in **Drucker**, in order to facilitate animation. It is for this reason that one of ordinary skill in the art would have been motivated to include geometrically coupling the first object to the second object in the object environment.

17. Claims 8 –13 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Szabo** in view of **Kross et al.** (hereinafter **Kross**, US 6,285,369).

18. Regarding claim 8, **Szabo** teaches a processor coupled to the storage system, wherein the processor is configured to open the first file [DerivedObject class] (See column 15, lines 7-8 "The derived object is an instance of the DerivedObject class. The

derived object instance is created when a user creates an object in a scene”), wherein the processor is configured to determine the second reference [pointer] in response to the first file (See column 15, lines 9 – 12 “Derived object 870 maintains a modifier stack. The modifier stack includes a list of modifiers...and a pointer to a base object 810.”), wherein the processor is configured to determine a value of an attribute from the second object [base object] in response to the first file, wherein the processor is configured to provide the second reference [pointer] to the storage system (See column 15, lines 9 – 12 “Derived object 870 maintains a modifier stack. The modifier stack includes a list of modifiers...and a pointer to a base object 810.”), wherein the processor is configured to receive the second file from the storage system (See column 16, lines 8-9 “The basic description is then provided to the object space modifier 820 in the modifier stack 880.”); wherein the processor is configured to open the second file [base object], and wherein the processor is configured to override a default value [modify the description] of the attribute from the second object with the value (See column 16, lines “Object space modifier 820 modifies this basic description to generate a modified description.”); wherein the second object is not stored in the first file. (See column 15, lines 9 – 14 “Derived object 870 maintains a modifier stack 880. The modifier stack 880 includes a list of modifiers..., a transform 830, and a pointer to a base object 810. The derived object 870 maintains a reference to base object 810, a reference to transform 830, and a list of modifiers.” Because the reference is stored in the file, the actual second object is not stored in the first file.)

Szabo fails to teach a storage system configured to store a first file describing a first object and a second file describing a second object, wherein the storage system is also configured to provide the first file in response to a first reference and configured to provide the second file in response to a second reference.

However, **Kross** teaches a storage system [notebook database] configured to store a first file [data file] describing a first object and a second file [data file] describing a second object, wherein the storage system is also configured to provide the first file [retrieves a current version of the object] in response to a first reference and configured to provide the second file in response to a second reference [retrieves a current version of the object]. (See column 6, lines 35 – 38 and 42-44 "...notebook database 252 maintains a record of the location of the object 342 (e.g., a location of a data file describing the object and a name of the object within that file)... Thus, whenever the note 445 is accessed, graphics view control 246 accesses the location of the object and retrieves a current version of the object 342.") One of ordinary skill in the art would have been motivated to combine the teaching of **Kross** with that of **Szabo** because they both are related to modifying objects, but by including the storage system of **Kross**, the actual object files are being stored in more of a persistent fashion to facilitate efficient operation of the system. It is for this reason that one of ordinary skill in the art would have been motivated to include a storage system configured to store a first file describing a first object and a second file describing a second object, wherein the storage system is also configured to provide the first file in response to a first reference and configured to provide the second file in response to a second reference.

19. Regarding claim 9, **Kross** additionally discloses the storage system comprises one of the group: network directory services, asset management system, database management system. (See column 4, lines 21 – 25 “As illustrated, electronic notebook 140 includes notebook database 25, main control logic 250, text view control logic 248...” these are describing aspects of a database management system.)

20. Regarding claim 10, **Szabo** additionally discloses the processor is also configured to modify the attribute [modifies the basic description] from the second object [base object] with a modified value [modifier] (See column 16, lines 7-11 “Specifically, base object 810 provides the basic description of the object. The basic description is then provided to the object space modifier 820 in the modifier stack 880. Object space modifier 820 modifies this basic description to generate a modified description.”); and wherein the first file [modifier stack] is modified to include the modified value [modifiers]. (See column 15, lines 10-11 “The modifier stack 880 includes a list of modifiers...”)

21. Regarding claim 11, **Szabo** additionally discloses the processor is also configured to geometrically manipulate [dynamically change] the first model and the second model. (See column 4, lines 57-60 “In addition, certain properties of a geometry object, such as the number of faces that are contained within a mesh representation of the geometry object, may dynamically change as the geometry object is passed up the stack.”)

22. Regarding claim 12, **Szabo** additionally discloses the processor is configured to determine the third reference [pointer] in response to the first file (See column 15, lines 9 – 12 “Derived object 870 maintains a modifier stack. The modifier stack includes a list of modifiers...and a pointer to a base object 810.”), wherein the processor is configured to determine a value of an attribute from the third object [base object] in response to the first file, wherein the processor is configured to provide the third reference [pointer] to the storage system (See column 15, lines 9 – 12 “Derived object 870 maintains a modifier stack. The modifier stack includes a list of modifiers...and a pointer to a base object 810.”), wherein the processor is configured to receive the third file from the storage system (See column 16, lines 8-9 “The basic description is then provided to the object space modifier 820 in the modifier stack 880.”); wherein the processor is configured to open the third file [base object], and wherein the processor is configured to override a default value [modify the description] of the attribute from the third object with the value (See column 16, lines “Object space modifier 820 modifies this basic description to generate a modified description.”).

Szabo fails to teach the storage system is also configured to store a third file describing a third object, and wherein the storage system is also configured to provide the third file in response to a third reference.

However, **Kross** teaches the storage system [notebook database] is also configured to store a third file [data file] describing a third object and wherein the storage system is also configured to provide the third file [retrieves a current version of

the object] in response to a third reference [retrieves a current version of the object].
(See column 6, lines 35 – 38 and 42-44 "...notebook database 252 maintains a record of the location of the object 342 (e.g., a location of a data file describing the object and a name of the object within that file)... Thus, whenever the note 445 is accessed, graphics view control 246 accesses the location of the object and retrieves a current version of the object 342.") One of ordinary skill in the art would have been motivated to combine the teaching of **Kross** with that of **Szabo** because they both are related to modifying objects, but by including the storage system of **Kross**, the actual object files are being stored in more of a persistent fashion to facilitate efficient operation of the system. It is for this reason that one of ordinary skill in the art would have been motivated to include the storage system is also configured to store a third file describing a third object, and wherein the storage system is also configured to provide the third file in response to a third reference.

23. Regarding claim 13, **Szabo** teaches the processor is configured to determine another instance of the second reference in response to the first file (See column 15, lines 7-9 "The derived object 870 is an instance of the DerivedObject class. The derived object instance is created when a user creates an object in a scene."), wherein the processor is configured to determined another instance of a value of an attribute [modifer] from the second object in response to the first file (See column 15, lines 9-10 "Derived object 870, maintains a modifier stack 880. The modifier stack 880 includes a list of modifiers..." Here, the value of the attribute is overridden based on that value of

the modifier in the modifier stack.), wherein the processor is configured to provide the second reference [pointer] to the storage system (See column 15, lines 9 – 12 “Derived object 870 maintains a modifier stack. The modifier stack includes a list of modifiers...and a pointer to a base object 810.”), wherein the processor is configured to receive the second file from the storage system (See column 16, lines 8-9 “The basic description is then provided to the object space modifier 820 in the modifier stack 880.”); wherein the processor is configured to open another instance of the second file [base object], and wherein the processor is configured to override a default value [modify the description] of the attribute from another instance of the third object with the value (See column 16, lines “Object space modifier 820 modifies this basic description to generate a modified description.”)

24. Claims 15 and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Szabo** as applied to claim 14 above and further in view of **Buxton**.

25. Regarding claim 15, **Szabo** teaches a computer program product substantially as claimed. **Szabo** fails to teach the server comprises one of the group: network directory, asset management system, database management system. However, **Buxton** teaches the server comprises one of the group: network directory, asset management system, database management system. (See column 10, lines 41-45 “An ISTORE is an OLE interface which supports the creation and management of structure storage objects. Such structured storage objects provide hierarchical storage of information

within a single file similar to a file system within a file.” This is similar to a network directory.) A person with ordinary skill in the art would have been motivated to combine the references because they both relate to the art of modifying data objects and they would have been motivated to include the server from **Buxton**, in order to maintain a persistent copy of the objects and modifiers rather than using a temporary stack, providing for a more efficient method. It is for this reason that one of ordinary skill in the art would have been motivated to include the server comprises one of the group: network directory, asset management system, database management system.

26. Regarding claim 16, **Szabo** teaches code that directs the processor to create a first file including the first object [derived object], the reference [pointer] to the second object [base object] in the server, and the modified value [modifiers] for the attribute; wherein the first file excludes the second object. (See column 15, lines 9 – 13 “Derived object 870 maintains a modifier stack 880. The modifier stack 880 includes a list of modifiers...and a pointer to a base object 810. The derived object 870 maintains a reference to a base object 810...and a list of modifiers.” Because the reference is stored in the file, the actual second object is not stored in the first file.)

Szabo fails to teach code that directs the processor to provide the first file to the server for storage.

However **Buxton** teaches code that directs the processor to provide the first file to the server for storage. (See column 11, lines 10-13 “In addition, these templates may be stored to facilitate efficient distribution of templates to others....” **Buxton** teaches

Art Unit: 2167

storing template files, whereas template files are not used in **Szabo**, rather a modifier stack is used.) A person with ordinary skill in the art would have been motivated to combine the references because they both relate to the art of modifying data objects and they would have been motivated to also include storing the reference and modified value, in order to facilitate a more efficient distribution of the reference and modifiers. It is for this reason that one of ordinary skill in the art would have been motivated to include code that directs the processor to provide the first file to the server for storage.

27. Claims 19 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Szabo** as applied to claim 17 above and further in view of **Buxton**.

28. Regarding claim 19, **Szabo** teaches a computer program substantially as claimed. **Szabo** fails to teach the plurality of attributes for the second object [object that is defined] include private attributes and public attributes; and wherein the public attributes includes the attribute. However, **Buxton** teaches the plurality of attributes for the second object include private attributes and public attributes; and wherein the public attributes includes the attribute. (See column 5, lines 48 – 54 “More particularly, during program design, a program developer can define objects in which all or some of the attributes and all or some of the related functions are considered ‘private’ or for use only by the object itself. Other data or functions can be declared ‘public’ or available for use by other programs.” In the instant application, as well as object oriented programming in general, the attributes that are modified have to be public in order for them to be

Art Unit: 2167

accessed.) A person with ordinary skill in the art would have been motivated to combine the references because they both relate to the art of modifying data objects and they would have been motivated to also include public and private attributes as disclosed in **Buxton**, because they provide for the ability to customize certain attributes while protecting others, leading to a flexible and efficient program. It is for this reason that one of ordinary skill in the art would have been motivated to include the plurality of attributes for the second object include private attributes and public attributes; and wherein the public attributes includes the attribute.

29. Regarding claim 20, **Szabo** additionally teaches the private attributes for the second object [base object] are the same for the instance of the second object and for the additional instance of the second object. (See column 16, line 7 – 8 “Specifically, base object 810 provides the basic description of the object.” The private attributes are the ones which are not able to be changed, and they are provided, as discussed here, when the description is provided for each object instantiated.)

Conclusion

30. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.


Gudmundson et al. (5,907,704) teaches creating reusable object containers by defining links among instantiated objects.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dennis L. Vautrot whose telephone number is 571-272-2184. The examiner can normally be reached on Monday-Friday 8:30-5:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cottingham can be reached on 571-272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Dv
1 September 2006


Primary Examiner
Art Unit 2167